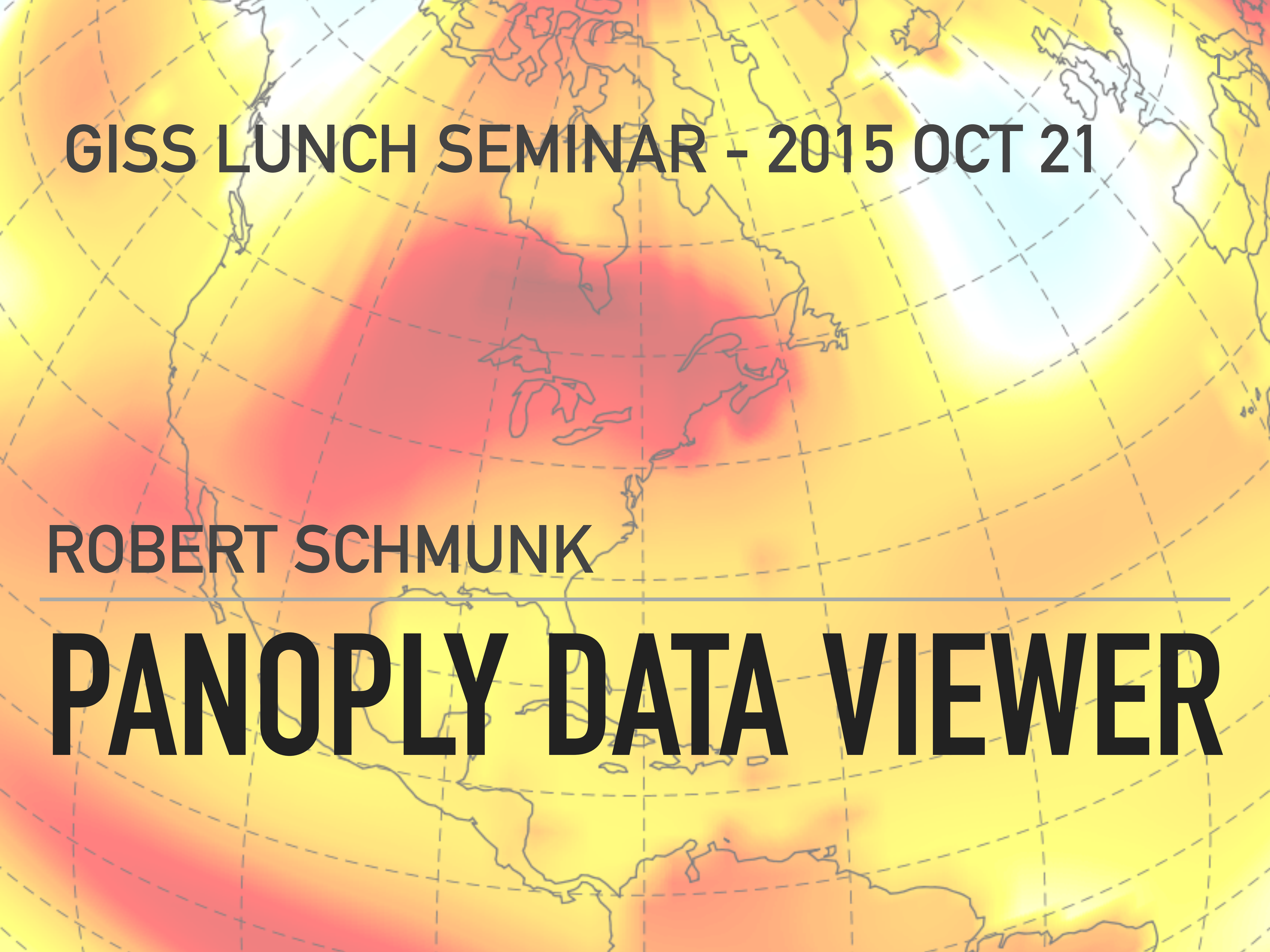
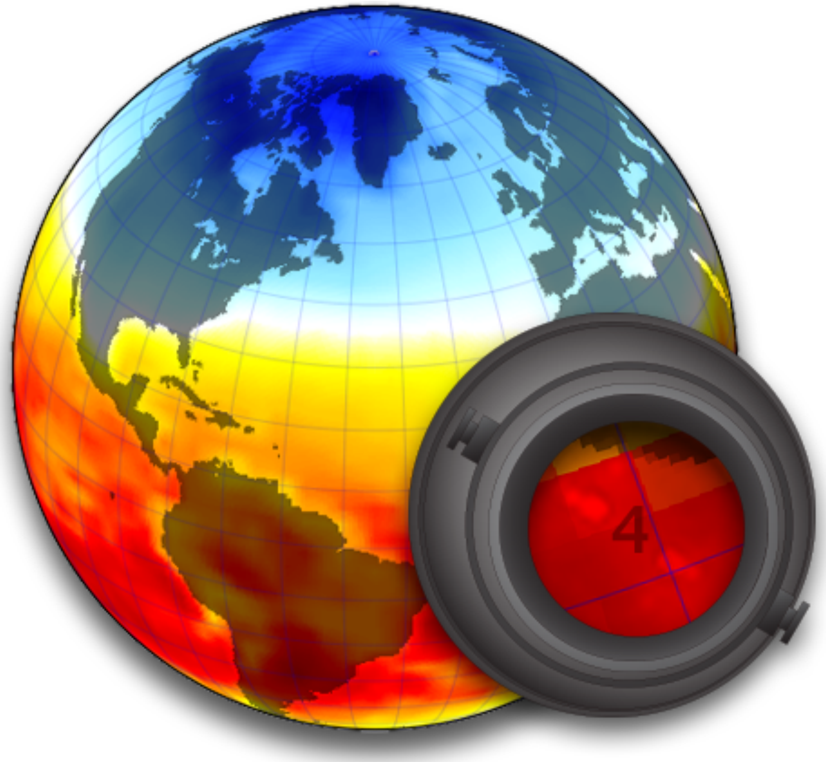


GISS LUNCH SEMINAR - 2015 OCT 21

ROBERT SCHMUNK

PANOPLY DATA VIEWER

PANOPLY



- GISS netCDF-HDF-GRIB Data Viewer
- "Push-button", interactive GUI application. Runs on Mac OS X, Windows, Linux, any desktop OS with Java 7 installed.
- Pronounced "pa-nuh-plee". Does not rhyme with monopoly.
- OED definition 3.a.: "A splendid or impressive array; fine or magnificent display; …".
- Latest update, version 4.3.1, released 2015 Sep 29.
- Download from www.giss.nasa.gov/tools/panoply

ORIGIN OF PANOPLY: WHY?

- GISS/CCSR EdGCM Project needed help with a data viewer/plotter that would run on both Mac OS X and on Windows.
- Educational users → viewer needs to be cheap or free, easy to install, and easy to use.
- Cross-platform on a budget → Java language (orig. Java 1.2; currently 1.7).
- Any similarities to 1990s Spyglass Transform a bonus.
- Development began Spring 2002. Version 1.0 released December 2002.
- Concurrent discovery of Unidata’s netCDF-Java (“NJ”) library. If EdGCM added netCDF export to model post-processing, then NJ library would handle data viewer’s interaction with dataset.

ORIGIN OF PANOPLY: 2002

- Panoply posted on GISS website for general download on release of ver. 1.0 in Dec 2002. Announced on giss-people mailing list Jan 2003.
- Soon saw use from users at other institutes, universities, labs, etc. And not just climate scientists.
- NetCDF software available in 2002-2003 definitely not as numerous as today and was often a bit clunky. For example:
- DOE/PNNL's ncbrowse - easy-to-install, but use of optional color tables difficult (if available at all) and no map projections.
- UCSD's ncview - not Java; required user to compile from source.
- Unidata's IDV - very new; in Java but required add'l install of Java 3D; not easy to figure out and performance was sluggish.

ORIGIN OF PANOPLY: VERSION 1.0

- Version 1.0 ***very*** focused on plotting lon-lat gridded GCM climate data.
- Optional color tables easy to use. Could be in Spyglass Transform's PAL, Adobe's ACT and Apple's CWC format.
- (Note: Default "panoply" and "panoply-diff" color tables date back to at least 1996, matching one used in Hansen et al. 1996 post-Pinatubo GISTEMP paper.)
- Offered choice of > 12 map projections: equirectangular, Mollweide, orthographic, Aitoff, azimuthal equal-area, etc.
- (Original map projection code lifted from GISS Mars24 app and expanded. Now includes > 125 projections.)
- Able to combine two variables via differencing.
- Did ***not*** allow for plotting data that was ***not*** on a lon-lat grid. "Generic 2D" plots not available for, *cough*, over 11 years. (Version 4.0, April 2014.)

ON-GOING DEVELOPMENT OF PANOPLY

- Version 1.1 released three weeks after version 1.0. Added continent outlines and more array combination methods.
- Ver. 1.2 - Zonal average line plots.
- Ver. 1.4 - Lat-vert plots. UDUNITS.
- Ver. 1.5 - Regional plots.
- Ver. 1.6 - PS and PDF output.
- Ver. 2.0 - Time-lat plots.
- Ver. 2.1 - Contour lines
- Ver. 2.2 - Able to read and plot rotated grid lon-lat data.
- Ver. 2.3 - Additional plot sizes, as long as they fit computer display size.
- Ver. 2.4 - Able to read and plot stereographic gridded data (e.g., WRF model output).
- Ver. 2.5 - Able to read irregular 2D lon-lat grids. Originally in context of regional oceanographic data but same scheme applies to satellite swath data.

ON-GOING DEVELOPMENT OF PANOPLY

- Ver. 2.6 - Vector plotting.
- Ver. 2.7 - netCDF version 4.
- Ver. 2.8 - Explicitly able to read and plot HDF data, iff able to interpret metadata.
- Ver 2.8.1 - Explicitly able to read and plot GRIB data.
- Ver. 2.8.2 - Export lon-lat plots as KMZ for use in Google Earth.
- Ver. 2.8.3 - JPEG and TIFF output.
- Ver. 2.9 - Export data in CSV and CDL text format from sources window.
- Ver. 3.0 - Bookmarking. Able to access remote THREDDS, OPENDAP catalogs. Animation export. Lon-vert plots.
- Ver. 3.2 - Able to read and plot some reduced horizontal grids.
- Ver. 4.0 - Generic 2D plots. Generic line plots. Lon-time and lat-time plots (Hovmöller diagrams).
- Ver. 4.2 - Time-vert plots.
- Ver. 4.2.2 - Able to change plot aspect ratio, i.e., taller or shorter.
- Ver. 4.3 - Fully vectorized PS and PDF output.

UNIDATA'S NETCDF-JAVA (NJ) LIBRARY

- NJ handles all interaction with dataset. Data viewer only has to do graphics and GUI.
- Panoply 1.0 performed own analysis to detect longitude and latitude coordinates. Evventually dropped in favor of using NJ to do so.
- NJ has over time substantially expanded ability to detect coordinate systems: lon, lat, vert, and time dimensions; projected grids; etc. (But still not 100% success.)
- To do so, NJ must open dataset in "enhanced mode". This also means NJ will automatically unpack compressed data (scaling and offsets) and interpret out-of-range and missing data.
- In some cases, NJ may construct coordinate system based on specified parameters rather than on explicit coordinate variables.
- But all this requires metadata.

METADATA

- Understanding a dataset without accompanying documentation requires metadata – data about data – conforming to a recognized convention.
- netCDF-Java (NJ) includes interpreters for many “registered” conventions: some specific to a particular model (e.g., WRF) but others very general.
- When in doubt, use CF (Climate and Forecast) Conventions: cfconventions.org.
- Also helps to use standard_names for variables and UDUNITS.
- Historically, netCDF dataset creators have been good about including metadata conforming to some standard or using simple schemes that code can figure out.
- Typical problems: bad units (“C” ≠ “degree Celsius”), too little metadata, or too much.

METADATA

- NJ library also reads HDF datasets. Understanding HDF dataset organization is trickier, but creators are quickly getting better about metadata.
- (Note: netCDF 4 ≈ HDF 5, but not exactly =.)
- NJ also reads some other dataset formats, e.g., GRIB (WMO GRIdded Binary). Might require external GRIBTAB, but many built in.
- Other formats quietly supported.

OPTIONAL COLOR TABLES

- Panoply comes with ~ 100 color tables built in, many variants on the Color Brewer sequential and diverging definitions.
- Some additional ACT format color tables available on Panoply website.
- Large collection of compatible CPT format color tables on “CPT City” website: soliton.vm.bytemark.co.uk/pub/cpt-city/
- Also see NCL (NCAR Graphics) color table gallery for RGB format color tables: www.ncl.ucar.edu/Document/Graphics/color_table_gallery.shtml

ON-GOING DEVELOPMENT OF PANOPLY

- New features and new capabilities often in response to user requests and feedback.
- Can you add X?
- Why can’t Panoply plot my data?
- My dataset uses this other convention, gridding scheme, etc.
- Response time not guaranteed. Other technical and service limitations may apply. Offer not valid in some locations.

PANOPLY DOES NOT

- Many things Panoply cannot do, yet:
- Combining > 2 arrays; e.g., averaging 12 months to plot an annual average, merging multiple satellite swaths, or creating ensemble line plots.
- Plotting of scattered data points; e.g., station data, buoy data, etc.
- Overlaying vector data of one quantity on color data of a different quantity; e.g., wind vectors on a temperature field.
- Plotting area data not on a 2D grid; e.g., triangular mesh.
- Vector plots from azimuth and magnitude data.
- Trajectory plots.
- Etc.

SOME EXAMPLES

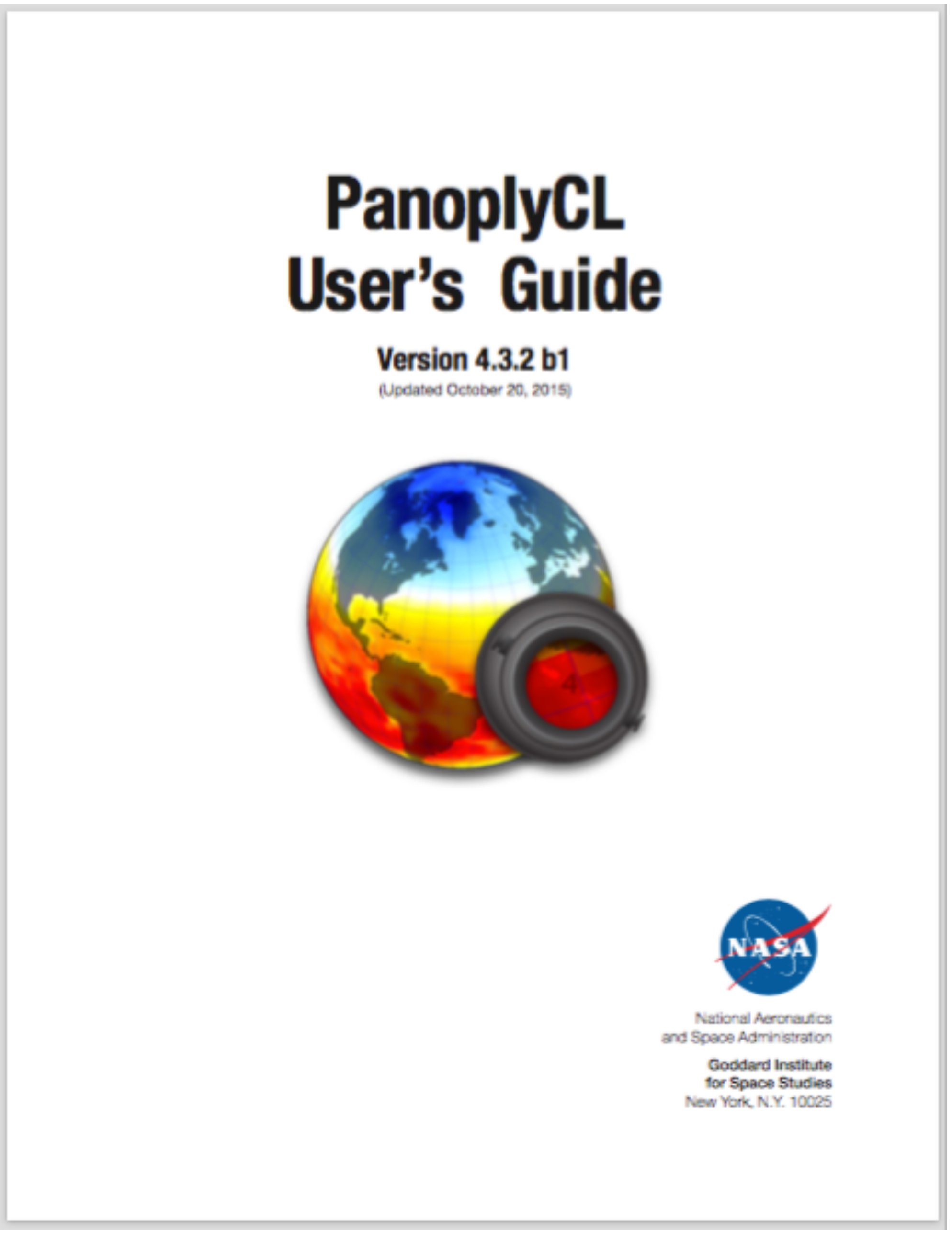
- Working with “Cartesian” gridded climate-model data.
- Working with projected grids.
- Working with “irregular” auxiliary grids.
- Loading remote catalog and accessing a dataset.

COMMAND-LINE PANOPLY?

- Many requests for non-GUI Panoply. Desired features include:
- Execute from console command line and create a plot based on an input file, i.e., a script.
- Able to re-use script at a later date, either to re-create a plot, to re-make a plot by tweaking parameters, or to re-make a plot with new data.
- Perhaps create multiple plots at one time.
- Perhaps run in server environment.
- Thus… PanoplyCL.

PANOPLYCL

- PanoplyCL currently in beta. Around two dozen “beta testers”, half at GISS.
- Some capabilities not yet (fully) coded, e.g., generic 2D plots and line plots.
- Documentation has some holes in it.
- Some aspects of API remain a moving target. Feedback would really help.
- Latest beta and user’s guide are at www.giss.nasa.gov/tools/panoply/…



PanoplyCL
User’s Guide

Version 4.3.2 b1
(Updated October 20, 2015)

National Aeronautics
and Space Administration
Goddard Institute
for Space Studies
New York, N.Y. 10025

PANOPLYCL SCRIPTING LANGUAGE

- Write scripts in JavaScript that can access exposed parts of PanoplyCL Java code.
- Java language can parse code in other languages using script engines.
- Oracle’s Java distribution includes either “Rhino” or “Nashorn” engine for parsing JavaScript.
- Third-party engines are available for other languages – e.g., “Jython” for Python – but PanoplyCL API is written with assumption you are using Javascript.
- Command-line call requires launching Java, telling it to execute PanoplyCL code, and also passing a script file name:

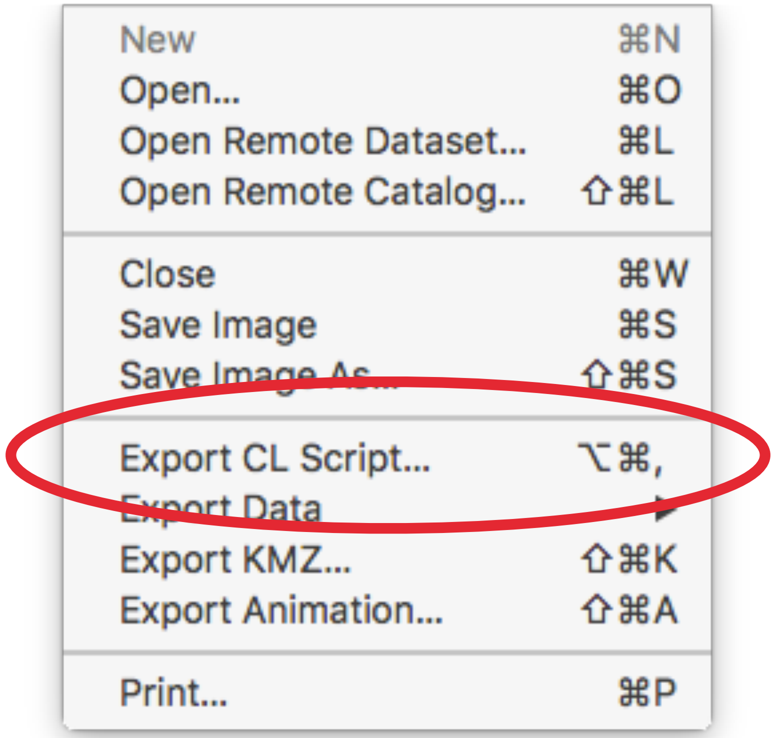
```
java -jar PanoplyCL.jar myscript.pjs
```

BASIC PANOPLYCL SCRIPT

```
// Script parsed by PanoplyCL to create and save a plot. Usage:
// java -jar PanoplyCL.jar “tas.pjs”

// Open dataset.
var ncdata1 = panoply.openDataset ( "/Users/rbs/data/modelE/pcmdi.ipcc4.giss_model_e_h.1pctto2x.run1.monthly.tas_A1.GISS3.1pctto2.nc" );

// Select variable.
var ncvar1 = ncdata1.getVariable ( "tas" );

// Create plot.
var myplot = panoply.createPlot ( "lonlat", ncvar1 );

// Use myplot.set (“parameter-name”, value ); to specify plot appearance settings.
// If setting not specified, then “factory default” value is applied.

// Specify plot settings.
myplot.set ( "size-factor", 200 );
myplot.set ( "title-text", "Surface Air Temperature" );

myplot.set ( "scale-min", 220. );
myplot.set ( "scale-max", 320. );

// Possibly ~ 40 myplot.set commands to specify plot appearance settings.
// . . .

// Variable #1 (tas), dim 1 (time) -- Set to step 2001 of 3000
myplot.setVarDimension ( 1, 1, 2001 );

// Save plot image to disk.
myplot.saveImage ( "PNG", "tas_in_pcmdi.ipcc4.giss_model_e_h.1pctt.png" );
```

GENERATING A SCRIPT

- No need to write a PanoplyCL script completely from scratch.
- Use Panoply to create a plot, then have it export a script that you can edit. Look for the “Export CL Script…” item in the File menu.
- Scripts can be complex, using common coding features as for-next loops, or math calculations to determine plot settings.
- For example: animating a map plot using an orthographic projection so that the Earth rotates from one frame to the next.



MORE COMPLEX PANOPLYCL SCRIPT

```
// java -jar PanoplyCL.jar “tas.pjs”

// Open dataset.
var ncdata1 = panoply.openDataset ( "/Users/rbs/data/modelE/pcmdi.ipcc4.giss_model_e_h.1pctto2x.run1.monthly.tas_A1.GISS3.1pctto2.nc" );

// Select variable.
var ncvar1 = ncdata1.getVariable ( "tas" );

// Create plot.
var myplot = panoply.createPlot ( "lonlat", ncvar1 );

// Specify plot settings.
myplot.set ( "size-factor", 110 );

myplot.set ( "proj-name", "Orthographic" );
myplot.set ( "proj-lat0", 45. );

// More myplot.set commands to specify plot appearance settings.
// . . .

var lonC = 0.

// Loop over the 3000 timesteps for variable TAS
for (var i = 1; i <= 3000; i++)
{
   // Select next timestep.
   myplot.setVarDimension ( 1, 1, i );

   // Rotate the projection a quarter degree from last frame.
   lonC += 0.25;
   myplot.set ( "proj-lon0", lonC );

   // Save plot image to disk. Javascript doesn’t have a sprintf function, so we have to format the number.
   var istr = "" + i;
   while (istr.length < 4) { istr = "0" + istr; }

   myplot.saveImage ("PNG", "TAS_" + istr + ".png");
}
```